[2022 Operating System] Progress Report

Group members: 407410003 常亦德、407410065 王宸潤、408410042 林靖紳

Introduction 80%

- Synchronization and locks will be introduced in class (mutual exclusion, progress, bound-waiting)
- We pay more attention on what is spinlock and its efficacy

Spinlock - method 60%

- We implement known spinlock method, inlcuding pthread_spin_lock in <pthread.h>, ticket-lock and qspinlock

```
// pthread_spin_lock()
pthread_spinlock_t spin;

void spin_init(){
    pthread_spin_init(&spin, PTHREAD_PROCESS_PRIVATE);
}
void spin_lock(){
    pthread_spin_lock(&spin);
}
void spin_unlock(){
    pthread_spin_unlock(&spin);
}
```

```
// ticket_lock()
atomic_ulong global_ticket;
atomic_ulong service_ticket;

void spin_init(){
    atomic_store(&global_ticket, 0);
    atomic_store(&service_ticket, 0);
}
void spin_lock(){
    unsigned long local_ticket = atomic_fetch_add(&global_ticket, 1);
    while (local_ticket != service_ticket)
        asm("pause");
}
void spin_unlock(){
    atomic_fetch_add(&service_ticket, 1);
}
```

Analysis 20%

- Each spinlock's locks per second (fix time)
- Spinlock v.s. Semphore and Mutex
  - on [single / muiltple] processes
  - in critical section is [arithmetic or FILE I/O]

Conclusion 10%

- Compare which spinlock method is better
  - (expected result: qspinlock will be the best)
- Where spinlock will > semphore and mutex
  - (we want explain why busy-waiting still alive)
- Where semaphore and mutex > spinlock